

Data Integration using Talend

Data Source: we will use “Orderline.csv” and “Product.csv” in this assignment. There are three columns: “OrderID”, “ProductID” and “Quantity” in Orderline data set and three columns: “ProductID”, “ProductName” and “Price” in the Product data set. The matched key column is “ProductID”.

Task 1: Sort and Filter

a. Import Product.csv into the Talend data studio job.



Job Task1 0.1

Product

8 rows in 0.09s
86.96 rows/s
row1 (Main)

tLogRow_1

Designer | Code

Job(Task1 0.1) Contexts(Task1) Component Run (Job Task1)

Job Task1

Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run

Execution

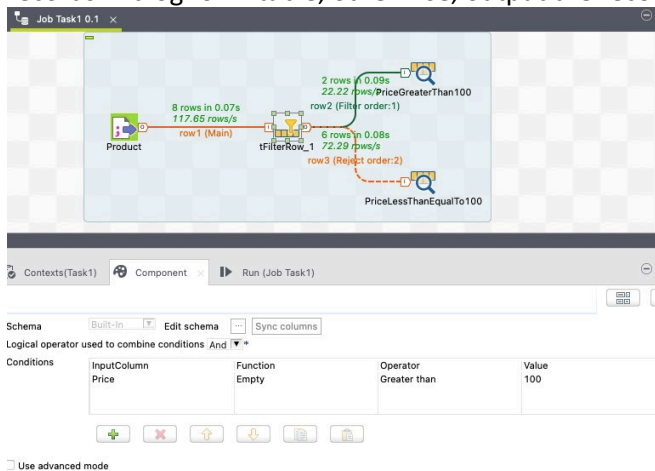
Run Kill Clear

Starting job Task1 at 17:23 28/07/2021.
[statistics] connecting to socket on port 4010
[statistics] connected

ProductID	ProductName	Price
101	DELL E5300 Laptop	489.98
102	Apple Laptop	988.72
103	Printer	59.0
104	Desk	85.98
105	Office Chair	55.99
106	Stapler	15.88
107	Index Divider	5.99
108	Shredder	74.99

[statistics] disconnected
Job Task1 ended at 17:23 28/07/2021. [Exit code = 0]

b. Filter the records from Product based on the value of Price. If the price is greater than 100, output the records in tLogRow1 table, otherwise, output the records in tLogRow2 table.



Job Task1 0.1

Product

8 rows in 0.07s
117.66 rows/s
row1 (Main)

tFilterRow_1

2 rows in 0.09s
22.22 rows/s/PriceGreaterThanOrEqualTo100
row2 (Filter order:1)

tLogRow_1

6 rows in 0.08s
72.29 rows/s
row3 (Reject order:2)

tLogRow_2

2 rows in 0.09s
22.22 rows/s/PriceLessThan100

Contexts(Task1) Component Run (Job Task1)

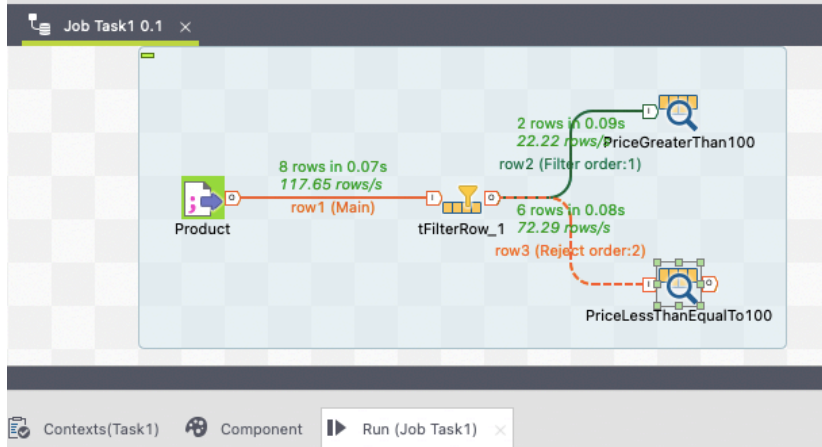
Schema Built-in Edit schema Sync columns

Logical operator used to combine conditions And

Conditions	InputColumn	Function	Operator	Value
	Price	Empty	Greater than	100

Use advanced mode

c. Take a screenshot of the executed job and paste it here; Copy and paste both table results from b here.



Execution

Run Kill Clear

ProductID	ProductName	Price
101	DELL E5300 Laptop	489.98
102	Apple Laptop	988.72

PriceLessThanOrEqualTo100			
ProductID	ProductName	Price	errorMessage
103	Printer	59.0	Price.compareTo(100) > 0 failed
104	Desk	85.98	Price.compareTo(100) > 0 failed
105	Office Chair	55.99	Price.compareTo(100) > 0 failed
106	Stapler	15.88	Price.compareTo(100) > 0 failed
107	Index Divider	5.99	Price.compareTo(100) > 0 failed
108	Shredder	74.99	Price.compareTo(100) > 0 failed

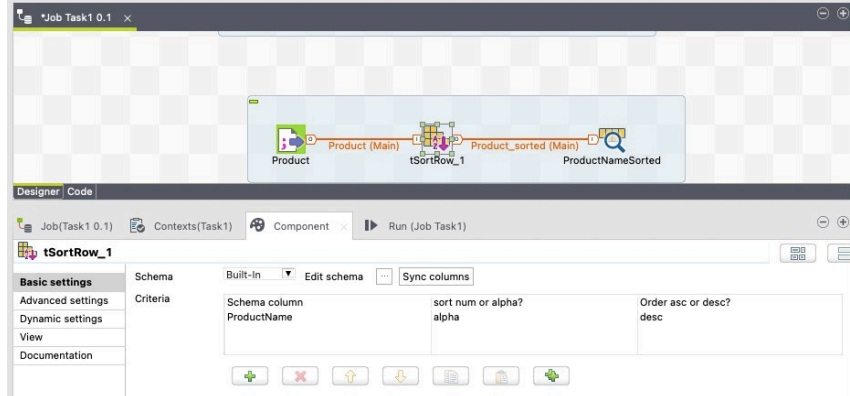
[statistics] disconnected

Job Task1 ended at 23:29 28/07/2021. [Exit code = 0]

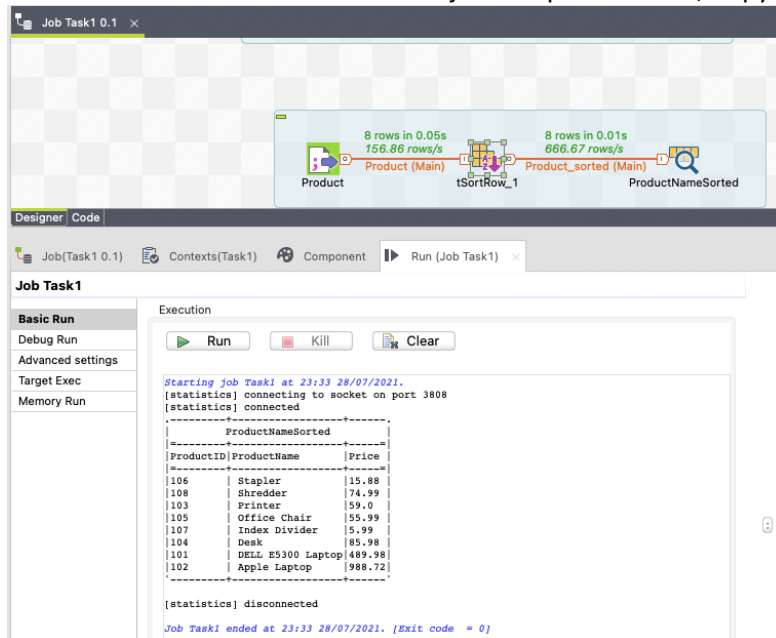
PriceGreaterThan100		
ProductID	ProductName	Price
101	DELL E5300 Laptop	489.98
102	Apple Laptop	988.72

PriceLessThanOrEqualTo100			
ProductID	ProductName	Price	errorMessage
103	Printer	59.0	Price.compareTo(100) > 0 failed
104	Desk	85.98	Price.compareTo(100) > 0 failed
105	Office Chair	55.99	Price.compareTo(100) > 0 failed
106	Stapler	15.88	Price.compareTo(100) > 0 failed
107	Index Divider	5.99	Price.compareTo(100) > 0 failed
108	Shredder	74.99	Price.compareTo(100) > 0 failed

d. Sort the records from Product by Product name in descending order. Output the results in tLogRow3 table.



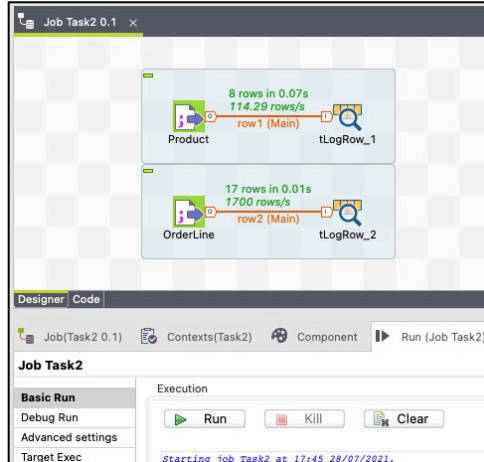
e. Take a screenshot of the executed job and paste it here; Copy and paste the table result from d here.



ProductSorted		
ProductID	ProductName	Price
106	Stapler	15.88
108	Shredder	74.99
103	Printer	59.0
105	Office Chair	55.99
107	Index Divider	5.99
104	Desk	85.98
101	DELL E5300 Laptop	489.98
102	Apple Laptop	988.72

Task 2: Inner Join and Integration

a. Import Product.csv and Orderline.csv into the Talend data studio job.



tLogRow_1		
ProductID	ProductName	Price
101	DELL E5300 Laptop	489.98
102	Apple Laptop	988.72
103	Printer	59.0
104	Desk	85.98
105	Office Chair	55.99
106	Stapler	15.88
107	Index Divider	5.99
108	Shredder	74.99

tLogRow_2		
OrderID	ProductID	Quantity
1001	101	2
1001	102	2
1001	104	1
1002	103	5
1003	103	3
1004	106	2
1004	107	2
1005	104	4
1006	104	1
1006	105	2
1006	107	2
1007	102	2
1008	103	3
1008	106	3
1009	104	2
1009	107	3
1010	105	10

'-----+-----+-----'

b. In one tLogRow table, output a table containing the matched records from Product and Orderline, with columns "ProductID", "ProductName", "Price" from Product, and columns "OrderID", "Quantity" from Orderline.

c. In another tLogRow table, output a table containing Product records that have not been purchased in any of the orders. Include "ProductID", "ProductName", "Price" from Product, and "OrderID", "Quantity" from Orderline in the output table.

d. Take a screenshot of the executed job and paste it here; Copy and paste both table results from b and c here.

The screenshot displays a data processing job named "Job Task2 0.1". The top section shows a flow diagram with inputs "Product" and "OrderLine" feeding into a "tmap_1" component. The "Product" input is labeled "8 rows in 0.01s 1333.33 rows/s Product (Main)". The "OrderLine" input is labeled "17 rows in 0.09s 184.78 rows/s OrderLine (Lookup)". The output of "tmap_1" is split into two paths: "output_b (Main order:1)" leading to "ProductsPurchased" (17 rows in 0.02s 809.52 rows/s) and "output_c (Main order:2)" leading to "ProductsNotPurchased" (1 rows in 0.02s 47.62 rows/s).

The bottom section shows the "Execution" results for "Job Task2". It includes a table of product data and a table for "ProductsNotPurchased".

ProductID	ProductName	Price	OrderID	Quantity
103	Printer	59.0	1008	3
104	Desk	85.98	1001	1
104	Desk	85.98	1005	4
104	Desk	85.98	1006	1
104	Desk	85.98	1009	2
105	Office Chair	55.99	1006	2
105	Office Chair	55.99	1010	10
106	Stapler	15.88	1004	2
106	Stapler	15.88	1008	3
107	Index Divider	5.99	1004	2
107	Index Divider	5.99	1006	2
107	Index Divider	5.99	1009	3

ProductID	ProductName	Price	OrderID	Quantity
108	Shredder	74.99	null	null

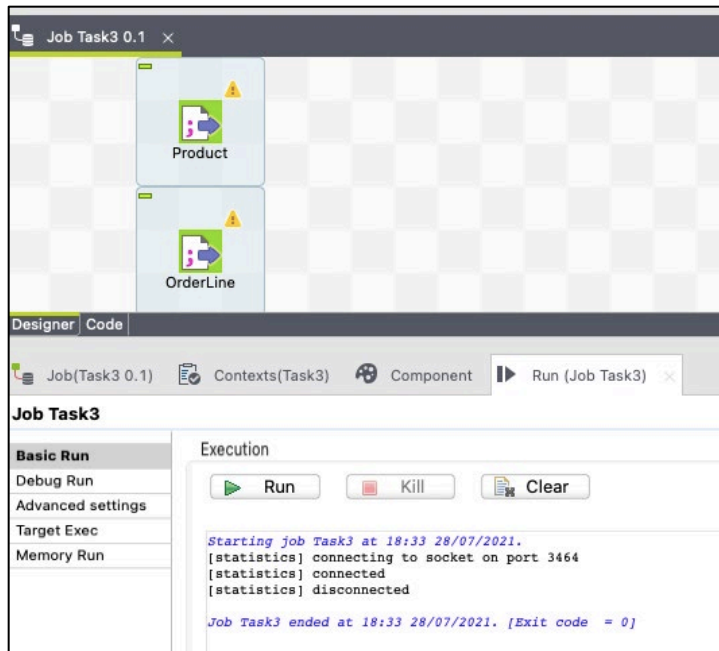
[statistics] disconnected
Job Task2 ended at 23:37 28/07/2021. [Exit code = 0]

ProductsPurchased				
ProductID	ProductName	Price	OrderID	Quantity
101	DELL E5300 Laptop	489.98	1001	2
102	Apple Laptop	988.72	1001	2
102	Apple Laptop	988.72	1007	2
103	Printer	59.0	1002	5
103	Printer	59.0	1003	3
103	Printer	59.0	1008	3
104	Desk	85.98	1001	1
104	Desk	85.98	1005	4
104	Desk	85.98	1006	1
104	Desk	85.98	1009	2
105	Office Chair	55.99	1006	2
105	Office Chair	55.99	1010	10
106	Stapler	15.88	1004	2
106	Stapler	15.88	1008	3
107	Index Divider	5.99	1004	2
107	Index Divider	5.99	1006	2
107	Index Divider	5.99	1009	3

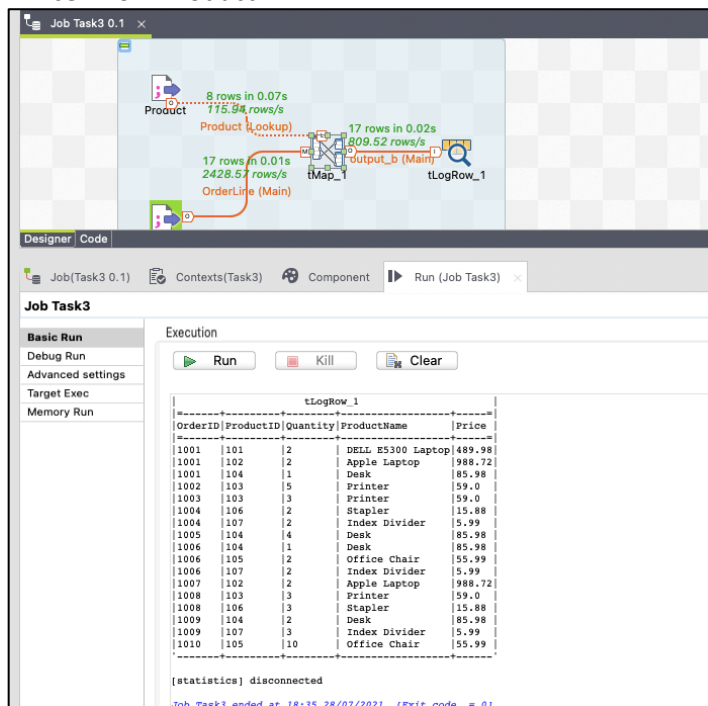
ProductsNotPurchased				
ProductName	ProductID	Price	OrderID	Quantity
Shredder	108	74.99	null	null

Task 3: Left Join and Aggregation

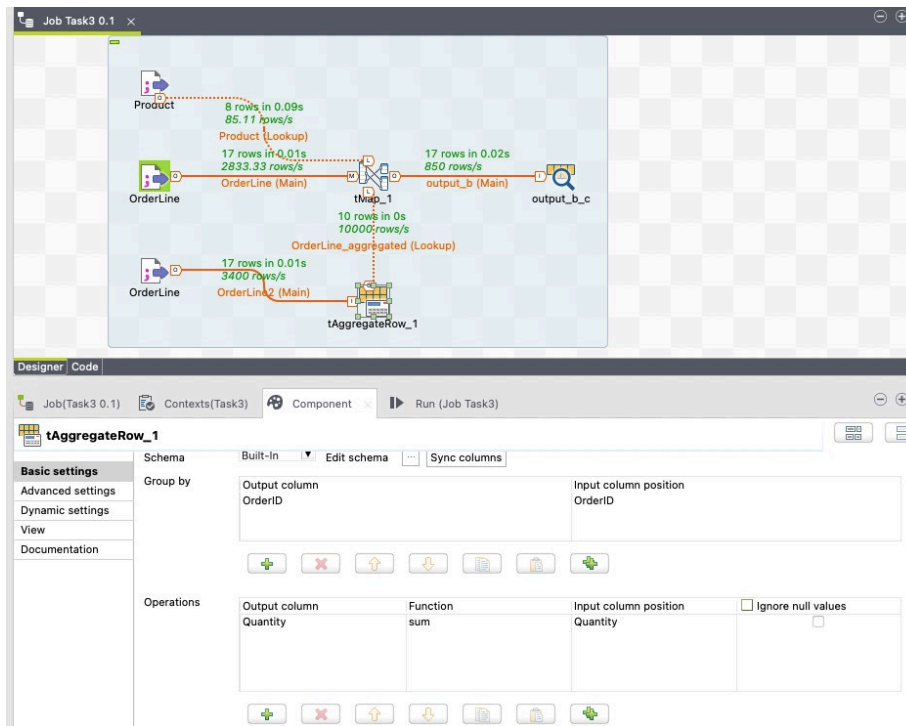
a. Import Product.csv and Orderline.csv into the Talend data studio job.



b. In the tLogRow table, output all records from Orderline and their corresponding records from Product, with columns “OrderID”, “ProductID”, “Quantity” from Orderline, and “ProductName” and “Price” from Product.



c. Aggregate the OverallQuantitybyOrder by summing the quantities from the same order (Order ID) and add OverallQuantitybyOrder further into the output table from b.



d. Take a screenshot of the executed job and paste it here; Copy and paste the table results from c here.

The screenshot shows the Job Task Designer interface with the execution results pane open. The top pane displays the same data flow diagram as in the previous screenshot. The bottom pane shows the execution results for the job.

Job Task3 Execution Results:

Run Kill Clear

1006	104	1	Desk	85.98	5
1006	105	2	Office Chair	55.99	5
1006	107	2	Index Divider	5.99	5
1007	102	2	Apple Laptop	988.72	2
1008	103	3	Printer	59.0	6
1008	106	3	Stapler	15.88	6
1009	104	2	Desk	85.98	5
1009	107	3	Index Divider	5.99	5
1010	105	10	Office Chair	55.99	10

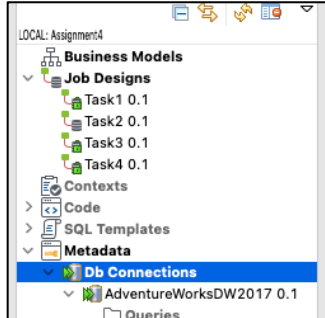
[statistics] disconnected

Job Task3 ended at 23:39 28/07/2021. [Exit code = 0]

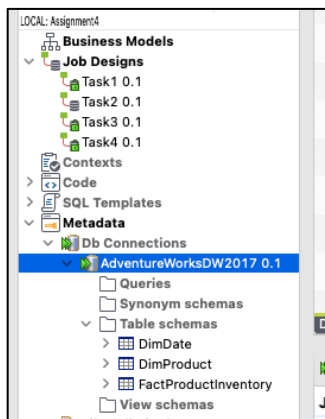
output_b_c						
OrderID	ProductID	Quantity	ProductName	Price	OverallQuantity	byOrder
1001	101	2	DELL E5300 Laptop	489.98	5	
1001	102	2	Apple Laptop	988.72	5	
1001	104	1	Desk	85.98	5	
1002	103	5	Printer	59.0	5	
1003	103	3	Printer	59.0	3	
1004	106	2	Stapler	15.88	4	
1004	107	2	Index Divider	5.99	4	
1005	104	4	Desk	85.98	4	
1006	104	1	Desk	85.98	5	
1006	105	2	Office Chair	55.99	5	
1006	107	2	Index Divider	5.99	5	
1007	102	2	Apple Laptop	988.72	2	
1008	103	3	Printer	59.0	6	
1008	106	3	Stapler	15.88	6	
1009	104	2	Desk	85.98	5	
1009	107	3	Index Divider	5.99	5	
1010	105	10	Office Chair	55.99	10	

Task 4: Data Warehouse

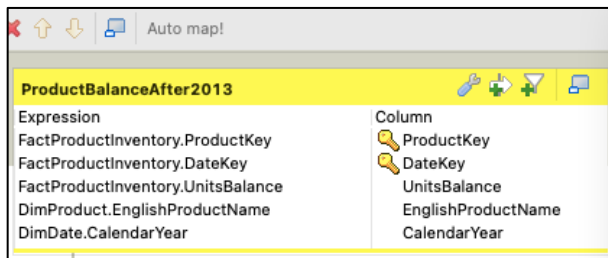
a. Use metadata to connect to Microsoft SQL server and the database: "AdventureWorksDW2017";



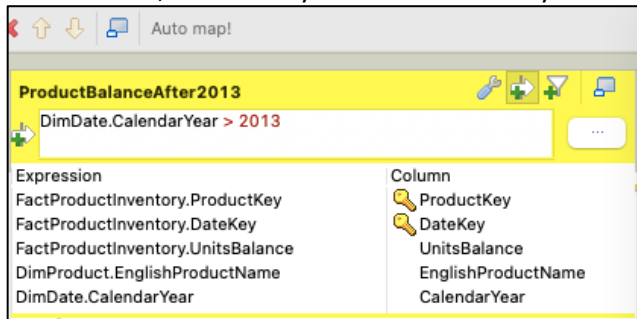
b. Import "DimProduct", "DimDate" and "FactProductInventory" tables into the job.



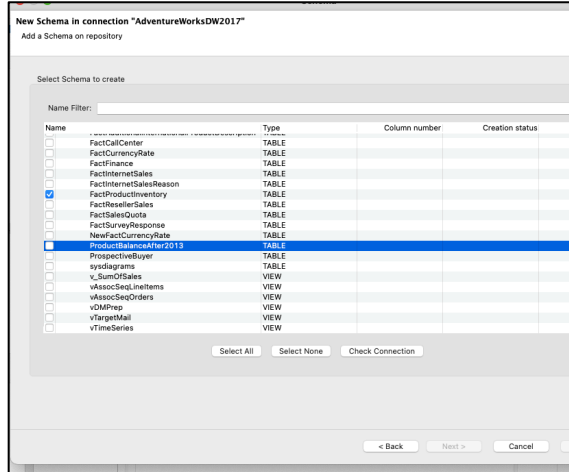
c. Match the key values in these tables, and generate a output table "ProductBalanceAfter2013" with the matching records and columns "ProductKey", "DateKey" and "UnitsBalance" from FactProductInventory table, "EnglishProductName" from DimProduct table, and "CalendarYear" from DimDate table.



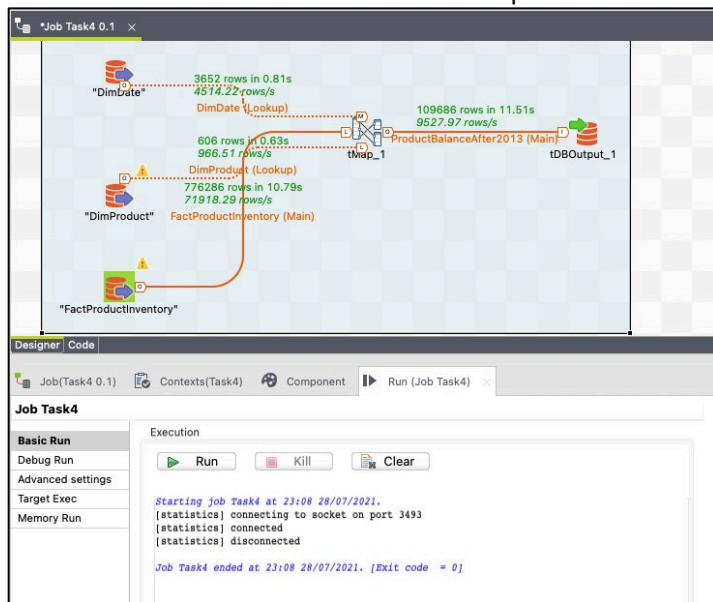
d. Set a filter, so that only records later than year 2013 are included in this output table.



e. Export this table to the database: “AdventureWorksDW2017”.



f. Take a screenshot of the executed job and paste it here. Use query to check the first 10 records from the table “ProductBalanceAfter2013” and paste it here.



Run Cancel Disconnect Change Connection AdventureWorksDW... Explain Enable

```

1 SELECT TOP(10) *
2 FROM ProductBalanceAfter2013;

```

Results Messages

	ProductKey	DateKey	UnitsBalance	EnglishProductName	CalendarYear
1	1	20140101	875	Adjustable Race	2014
2	1	20140102	875	Adjustable Race	2014
3	1	20140103	875	Adjustable Race	2014
4	1	20140104	875	Adjustable Race	2014
5	1	20140105	875	Adjustable Race	2014
6	1	20140106	875	Adjustable Race	2014
7	1	20140107	875	Adjustable Race	2014
8	1	20140108	875	Adjustable Race	2014
9	1	20140109	875	Adjustable Race	2014
10	1	20140110	875	Adjustable Race	2014